

1.5 c)

Tenjamos restringir o tempo de execuções em 30 %.

16 > - 30% = 7 Qual o tempo de solução que obtemos com isto?

$$\text{Tempo de CPU}_{P_1} = 10 - 3 = 7 \rightarrow$$

$$\text{Tempo de CPU}_{P_2} = 10 - 3 = 7 \rightarrow$$

$$\text{Tempo de CPU}_{P_3} = 10 - 3 = 7 \rightarrow$$

$$\frac{\text{CPU}_{P_1}}{10} = \frac{\text{CPU}_{P_1} \text{ depois}}{10} = \frac{10 - 3}{10} = 0,7 \rightarrow$$

$$\frac{\text{CPU}_{P_2}}{10} = \frac{\text{CPU}_{P_2} \text{ depois}}{10} = \frac{10 - 3}{10} = 0,7 \rightarrow$$

$$\frac{\text{CPU}_{P_3}}{10} = \frac{\text{CPU}_{P_3} \text{ depois}}{10} = \frac{10 - 3}{10} = 0,7 \rightarrow$$

(P3)

ePTI depois

$$1,2 \rightarrow 100$$

$$1,0 \rightarrow 100$$

$$0,8 \rightarrow 100$$

$$0,6 \rightarrow 100$$

$$0,4 \rightarrow 100$$

$$0,2 \rightarrow 100$$

$$0,0 \rightarrow 100$$

$$-0,2 \rightarrow 100$$

$$-0,4 \rightarrow 100$$

$$-0,6 \rightarrow 100$$

$$-0,8 \rightarrow 100$$

$$-1,0 \rightarrow 100$$

$$-1,2 \rightarrow 100$$

$$-1,4 \rightarrow 100$$

$$-1,6 \rightarrow 100$$

$$-1,8 \rightarrow 100$$

$$-2,0 \rightarrow 100$$

$$-2,2 \rightarrow 100$$

$$-2,4 \rightarrow 100$$

$$-2,6 \rightarrow 100$$

$$-2,8 \rightarrow 100$$

$$-3,0 \rightarrow 100$$

$$-3,2 \rightarrow 100$$

$$-3,4 \rightarrow 100$$

$$-3,6 \rightarrow 100$$

$$-3,8 \rightarrow 100$$

$$-4,0 \rightarrow 100$$

$$-4,2 \rightarrow 100$$

$$-4,4 \rightarrow 100$$

$$-4,6 \rightarrow 100$$

$$-4,8 \rightarrow 100$$

$$-5,0 \rightarrow 100$$

$$-5,2 \rightarrow 100$$

$$-5,4 \rightarrow 100$$

$$-5,6 \rightarrow 100$$

$$-5,8 \rightarrow 100$$

$$-6,0 \rightarrow 100$$

$$-6,2 \rightarrow 100$$

$$-6,4 \rightarrow 100$$

$$-6,6 \rightarrow 100$$

$$-6,8 \rightarrow 100$$

$$-7,0 \rightarrow 100$$

$$-7,2 \rightarrow 100$$

$$-7,4 \rightarrow 100$$

$$-7,6 \rightarrow 100$$

$$-7,8 \rightarrow 100$$

$$-8,0 \rightarrow 100$$

$$-8,2 \rightarrow 100$$

$$-8,4 \rightarrow 100$$

$$-8,6 \rightarrow 100$$

$$-8,8 \rightarrow 100$$

$$-9,0 \rightarrow 100$$

$$-9,2 \rightarrow 100$$

$$-9,4 \rightarrow 100$$

$$-9,6 \rightarrow 100$$

$$-9,8 \rightarrow 100$$

$$-10,0 \rightarrow 100$$

$$-10,2 \rightarrow 100$$

$$-10,4 \rightarrow 100$$

$$-10,6 \rightarrow 100$$

$$-10,8 \rightarrow 100$$

$$-11,0 \rightarrow 100$$

$$-11,2 \rightarrow 100$$

$$-11,4 \rightarrow 100$$

$$-11,6 \rightarrow 100$$

$$-11,8 \rightarrow 100$$

$$-12,0 \rightarrow 100$$

$$-12,2 \rightarrow 100$$

$$-12,4 \rightarrow 100$$

$$-12,6 \rightarrow 100$$

$$-12,8 \rightarrow 100$$

$$-13,0 \rightarrow 100$$

$$-13,2 \rightarrow 100$$

$$-13,4 \rightarrow 100$$

$$-13,6 \rightarrow 100$$

$$-13,8 \rightarrow 100$$

$$-14,0 \rightarrow 100$$

$$-14,2 \rightarrow 100$$

$$-14,4 \rightarrow 100$$

$$-14,6 \rightarrow 100$$

$$-14,8 \rightarrow 100$$

$$-15,0 \rightarrow 100$$

$$-15,2 \rightarrow 100$$

$$-15,4 \rightarrow 100$$

$$-15,6 \rightarrow 100$$

$$-15,8 \rightarrow 100$$

$$-16,0 \rightarrow 100$$

$$-16,2 \rightarrow 100$$

$$-16,4 \rightarrow 100$$

$$-16,6 \rightarrow 100$$

$$-16,8 \rightarrow 100$$

$$-17,0 \rightarrow 100$$

$$-17,2 \rightarrow 100$$

$$-17,4 \rightarrow 100$$

$$-17,6 \rightarrow 100$$

$$-17,8 \rightarrow 100$$

$$-18,0 \rightarrow 100$$

$$-18,2 \rightarrow 100$$

$$-18,4 \rightarrow 100$$

$$-18,6 \rightarrow 100$$

$$-18,8 \rightarrow 100$$

$$-19,0 \rightarrow 100$$

$$-19,2 \rightarrow 100$$

$$-19,4 \rightarrow 100$$

$$-19,6 \rightarrow 100$$

$$-19,8 \rightarrow 100$$

$$-20,0 \rightarrow 100$$

$$-20,2 \rightarrow 100$$

$$-20,4 \rightarrow 100$$

$$-20,6 \rightarrow 100$$

$$-20,8 \rightarrow 100$$

$$-21,0 \rightarrow 100$$

$$-21,2 \rightarrow 100$$

$$-21,4 \rightarrow 100$$

$$-21,6 \rightarrow 100$$

$$-21,8 \rightarrow 100$$

$$-22,0 \rightarrow 100$$

$$-22,2 \rightarrow 100$$

$$-22,4 \rightarrow 100$$

$$-22,6 \rightarrow 100$$

$$-22,8 \rightarrow 100$$

$$-23,0 \rightarrow 100$$

$$-23,2 \rightarrow 100$$

$$-23,4 \rightarrow 100$$

$$-23,6 \rightarrow 100$$

$$-23,8 \rightarrow 100$$

$$-24,0 \rightarrow 100$$

$$-24,2 \rightarrow 100$$

$$-24,4 \rightarrow 100$$

$$-24,6 \rightarrow 100$$

$$-24,8 \rightarrow 100$$

$$-25,0 \rightarrow 100$$

$$-25,2 \rightarrow 100$$

$$-25,4 \rightarrow 100$$

$$-25,6 \rightarrow 100$$

$$-25,8 \rightarrow 100$$

$$-26,0 \rightarrow 100$$

$$-26,2 \rightarrow 100$$

$$-26,4 \rightarrow 100$$

$$-26,6 \rightarrow 100$$

$$-26,8 \rightarrow 100$$

$$-27,0 \rightarrow 100$$

$$-27,2 \rightarrow 100$$

$$-27,4 \rightarrow 100$$

$$-27,6 \rightarrow 100$$

$$-27,8 \rightarrow 100$$

$$-28,0 \rightarrow 100$$

$$-28,2 \rightarrow 100$$

$$-28,4 \rightarrow 100$$

$$-28,6 \rightarrow 100$$

$$-28,8 \rightarrow 100$$

$$-29,0 \rightarrow 100$$

$$-29,2 \rightarrow 100$$

$$-29,4 \rightarrow 100$$

$$-29,6 \rightarrow 100$$

$$-29,8 \rightarrow 100$$

$$-30,0 \rightarrow 100$$

$$-30,2 \rightarrow 100$$

$$-30,4 \rightarrow 100$$

$$-30,6 \rightarrow 100$$

$$-30,8 \rightarrow 100$$

$$-31,0 \rightarrow 100$$

$$-31,2 \rightarrow 100$$

$$-31,4 \rightarrow 100$$

$$-31,6 \rightarrow 100$$

$$-31,8 \rightarrow 100$$

$$-32,0 \rightarrow 100$$

$$-32,2 \rightarrow 100$$

$$-32,4 \rightarrow 100$$

$$-32,6 \rightarrow 100$$

$$-32,8 \rightarrow 100$$

$$-33,0 \rightarrow 100$$

$$-33,2 \rightarrow 100$$

$$-33,4 \rightarrow 100$$

$$-33,6 \rightarrow 100$$

$$-33,8 \rightarrow 100$$

$$-34,0 \rightarrow 100$$

$$-34,2 \rightarrow 100$$

$$-34,4 \rightarrow 100$$

$$-34,6 \rightarrow 100$$

$$-34,8 \rightarrow 100$$

$$-35,0 \rightarrow 100$$

$$-35,2 \rightarrow 100$$

$$-35,4 \rightarrow 100$$

$$-35,6 \rightarrow 100$$

$$-35,8 \rightarrow 100$$

$$-36,0 \rightarrow 100$$

$$-36,2 \rightarrow 100$$

$$-36,4 \rightarrow 100$$

$$-36,6 \rightarrow 100$$

$$-36,8 \rightarrow 100$$

$$-37,0 \rightarrow 100$$

$$-37,2 \rightarrow 100$$

$$-37,4 \rightarrow 100$$

$$-37,6 \rightarrow 100$$

$$-37,8 \rightarrow 100$$

$$-38,0 \rightarrow 100$$

$$-38,2 \rightarrow 100$$

$$-38,4 \rightarrow 100$$

$$-38,6 \rightarrow 100$$

$$-38,8 \rightarrow 100$$

$$-39,0 \rightarrow 100$$

$$-39,2 \rightarrow 100$$

$$-39,4 \rightarrow 100$$

$$-39,6 \rightarrow 100$$

$$-39,8 \rightarrow 100$$

$$-40,0 \rightarrow 100$$

$$-40,2 \rightarrow 100$$

$$-40,4 \rightarrow 100$$

$$-40,6 \rightarrow 100$$

$$-40,8 \rightarrow 100$$

$$-41,0 \rightarrow 100$$

$$-41,2 \rightarrow 100$$

$$-41,4 \rightarrow 10$$

$$\textcircled{P2} \quad \text{Total ePDU defor} = \frac{N_i \times \text{ePDU defor}}{\text{Fe defor}}$$

$$\text{ePDU time (anter)} = \frac{m^{\circ} \text{ int}}{\text{Fe (anter)}} \times \text{ePT (anter)}$$

$$\text{ePDU time (defor)} = \frac{m^{\circ} \text{ int}}{\text{Fe (defor)}} \times \text{ePT (defor)}$$

$$\Rightarrow \text{ePDU time (anter)} = \frac{m^{\circ} \text{ int} \times \text{ePT (anter)}}{m^{\circ} \text{ int} \times \text{ePT (defor)} \times \text{Fe (defor)}}$$

$$\frac{\text{Fe (anter)}}{\text{Fe (defor)}} = \frac{m^{\circ} \text{ int} \times \text{ePT (anter)}}{m^{\circ} \text{ int} \times \text{ePT (defor)} \times \text{ePDU time (defor)}}$$

$$\frac{2,5 \times 10^9}{\text{Fe defor}} = \frac{2,15 \times 10^{10}}{2,5 \times 10^{10}} \times \frac{1,0}{1,2} \times \frac{7}{10}$$

$$2,5 \times 10^9 = \frac{1,75 \times 10^{10}}{3,0 \times 10^{11}}$$

$$\frac{7,5 \times 10^{20}}{1,75 \times 10^{11}} = \text{Fe defor}$$

$$\text{Fe defor} = 4,3 \times 10^9 \text{ Hz} = 4,285 \text{ GHz}$$

$$\textcircled{P3} \quad \frac{4,0 \times 10^9}{\text{Fe defor}} = \frac{1,82 \times 10^{10} \times 2,2 \times 7}{1,82 \times 10^{10} \times 2,64 \times 10}$$

$$\text{Fe defor} = \frac{4,804 \times 10^{11}}{2,802 \times 10^{11}}$$

$$\text{Fe defor} = 1,859 \times 10^9 \text{ Hz}$$

$$\text{Fe defor} = 6,186 \text{ GHz}$$

1.6

Consider 2 implementaciones con los mismos números de instrucciones. An implementación poseen la dividida en 4 claves de acuerdo como

CPI

(claw A, B, C, D)

P1 tiene Fe.

2, 5 GHz

CPI's

1, 2, 3, 3

P2 tiene Fe. 3 GHz

CPI's

2, 2, 2, 2

Sabemos programas con instrucción invocadas

1, 0 x 10⁶ instrucciones.

Distribución por

10% claw A

20% claw B

50% claw C

20% claw D

¿Cuál implementación es más rápida?

Fe	CPI claw A	CPI claw B	CPI claw C	CPI claw D
P1 2,5 x 10 ⁹	1	2	3	2
P2 3 x 10 ⁹	2	2	1	2

Nº instrucciones	1 x 10 ⁶
claw A	claw B
10%	50%
0,1 x 10 ⁶	0,5 x 10 ⁶

¿Qué es CPI global para cada implementación?

$$CPI_{global} = \sum \text{% de ejecución} \times CPI_{cada}$$

$$CPI_{global} = (0,1 \times 10^6 \times 1) + (0,2 \times 10^6 \times 2) + (0,5 \times 10^6 \times 3) + (0,2 \times 10^6 \times 3)$$

$$= 0,1 \times 2,5 + 0,4 \times 5 + 1,5 \times 3 + 0,6 \times 3 =$$

$$= 2,6 \times 10^6 /$$

$$CPI_{global} = (0,1 \times 10^6 \times 2) + (0,2 \times 10^6 \times 2) + (0,5 \times 10^6 \times 2) + (0,2 \times 10^6 \times 2)$$

$$= 0,1 \times 3 + 0,4 \times 3 + 1,0 \times 3 + 0,4 \times 3$$

$$= 2 \times 10^6 /$$

1.6. b) Encontra-se o C.R. de relogar necessario para cada instalação com 2 implementações

$$N_{\text{celdas}} = N_{\text{implementações}} \times ePI$$

$$\begin{aligned} N_{\text{celdas}} &= 1 \times 10^6 \times 26 \times 2 \\ &= 26 \times 10^6 \end{aligned}$$

$$\begin{aligned} m_{\text{celdas}} &= 1 \times 10^6 \times 2,0 \times 10^{-6} \\ p_2 &= 2,0 \times 10^6 \end{aligned}$$

16. c) C.R.U Time da 2 implementação? $ePDU_{\text{max}} = \frac{\text{arriendo} \times (ePI)}{F_2}$

$$ePDU_{\text{max}} = \frac{1 \times 10^6}{2,5 \times 10^{-9}} \times (1 \times 0,1 \times 10^6) + (2 \times 0,2 \times 10^6) + (3 \times 0,5 \times 10^6) + (3 \times 0,2 \times 10^6)$$

$$ePDU_{\text{max}} P_2 = 1040 \text{ PUs}$$

$$ePDU_{\text{max}} P_1 = 1,04 \times 10^{-3} \Delta //$$

$$ePDU_{\text{mínimo}} = \frac{1 \times 10^6 \times 2 \times 10^6}{3 \times 10^9}$$

$$\begin{aligned} ePDU_{\text{mínimo}} P_2 &= 666,666 \\ &= 0,666 \times 10^{-3} // \end{aligned}$$

17. Considera-se que temos uma implementação de software que impacta na performance do sistema. A medida é feita para um programa complexo A que demora 1×10^{-9} s e um tempo de execução de 1 ms quando que o software que compõe B tem uma execução de 2×10^{-9} s. Tendo em conta que o tempo total é de 15 s. Encontra o e.P.I. para cada programa tendo em conta que o tempo total é de 1 ms.

Nº instalações	Tempo de execução	Tempo total de utilização	ePI
Conf. A	1×10^{-9}	1,15	$1 \text{ ms} = 1 \times 10^{-9}$
Conf. B	$1,2 \times 10^{-9}$	1,51	$1 \text{ ms} = 1 \text{ ms} \times 10^{-9}$

Já que temos $N_{\text{implementações}} = 2$

$$N_{\text{celdas}} = \text{arriendo} \times ePI$$

$$ePI = \frac{N_{\text{celdas}}}{\text{arriendo}}$$

$$ePI(A) = \frac{1}{1 \times 10^{-9}} = 1 \times 10^9$$

nº inst	Tam pulpa	Tam de número	ePI	+ poligo
Cont A	1×10^9	$1,11 \Delta$	$(1m) = 1 \times 10^{-9}$	$1/1$
cont B	$1,2 \times 10^9$	$1,5 \Delta$	$(1m) 2 \times 10^{-9} \Delta$	$1,25$
Cont C	6×10^8	$0,66 \Delta$		$1,1$

a) $ePI = ?$

$$ePI = \frac{n^o \text{ celas}}{N^o \text{ instancias}}$$

$$ePI_A =$$

$$+ \log_{10} = \frac{1}{t}$$

$$+ \log_{10} (\text{Cont A}) = \frac{1}{1 \times 10^9} = 1 \times 10^{-9} \text{ Hz}$$

$$+ \log_{10} (\text{Cont B}) = 1 \times 10^{-9} \text{ Hz}$$

$$\text{Total, ePI}_A = \frac{n^o \text{ inst}(A) \times ePI(A) \times 1}{\text{Foligio}(A)}$$

$$ePI(\text{Cont B}) = \frac{1,5 \times 1 \times 10^9}{1,2 \times 10^9}$$

$$\frac{1,1 \times 10^9}{1 \times 10^9} = ePI(A)$$

$$ePI(\text{Cont A}) = 1,1 \text{ celas} //$$

$$ePI(\text{Cont B}) = 1,25 \text{ celas} //$$

b)

$$\frac{+ \log_{10} (\text{Cont B})}{+ \log_{10} (\text{Cont A})} = \frac{n^o \text{ inst}(B) \times ePI(B)}{n^o \text{ inst}(A) \times ePI(A)} = \frac{1,2 \times 10^9 \times 1,25}{1 \times 10^9 \times 1,1} = 1,363 \text{ resp}$$

//

Também consta que o tam de escala é igual a $1,363 \text{ resp}$, B tem num. primos e logaritmo maior que A

② o mort compilare ! Quel est speedup de plus étendue

Compilare un réseau ~ A + B

$$CPUTime_{\text{total}} = \frac{n^{\text{mat}}(e) \times CPUTime_e}{\# \text{ Register}(e)}$$

$$CPUTime(\text{compl}) = \frac{6 \times 10^8 \times 1,1}{1 \times 10^9} = 0,66 \text{ s}$$

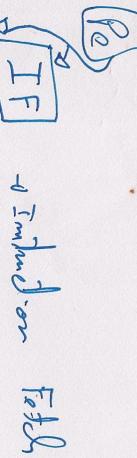
$$\text{speedup} = \frac{CPUTime_A}{CPUTime_B} = \frac{1,1}{0,66} = 1,667$$

Complexité est moins n'importe 1,67 fois
de plus compliqué A

$$\text{speedup} = \frac{CPUTime_B}{CPUTime_A} = \frac{1,1}{1,66} = 0,667$$

(la compilare est moins n'importe 2,273 fois que à compiler B)

1.9



→ Instruction Fetch



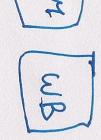
Instruction Decoder
Le o contenido de memoria
interpreta la instrucción

[ID] → ALU - operación Logica o aritmética

[Mem DM] manda los datos

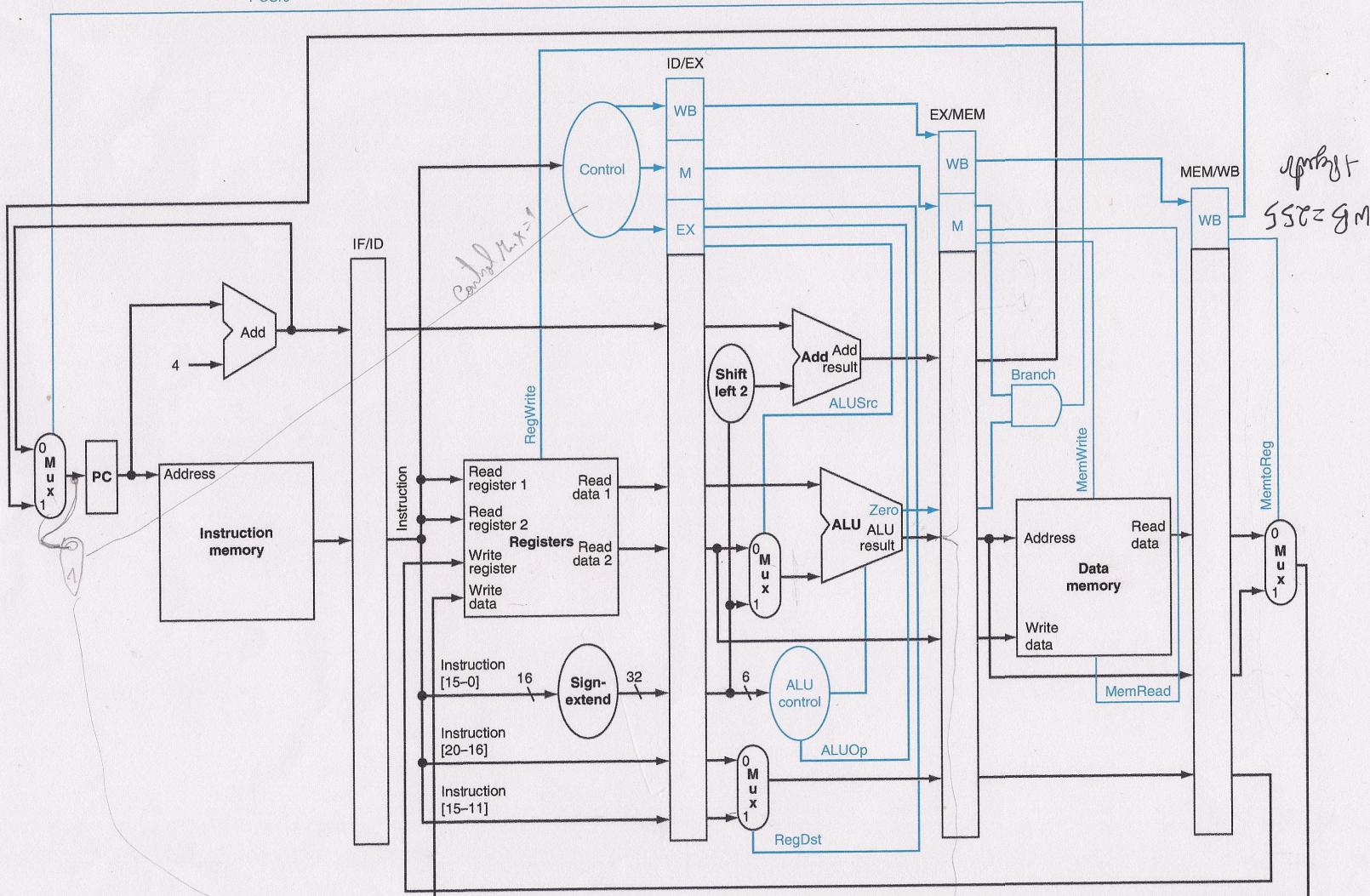
[WB] manda datos en regreso

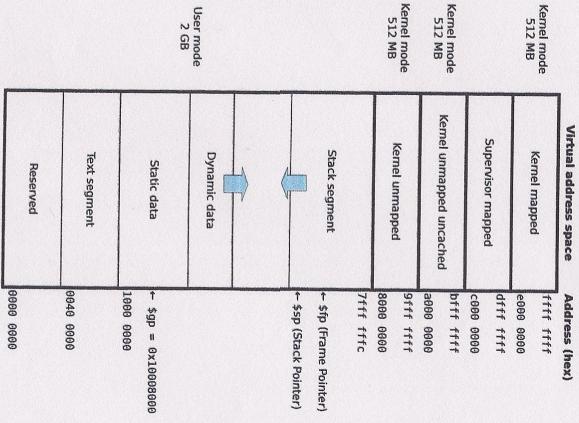
Mismo problema k instrucción sea ejecutable p' 15



No hay un bucle de memoria para la fase

PCSrc





ASCII / 2008

Miguel Barrio

Green 008000

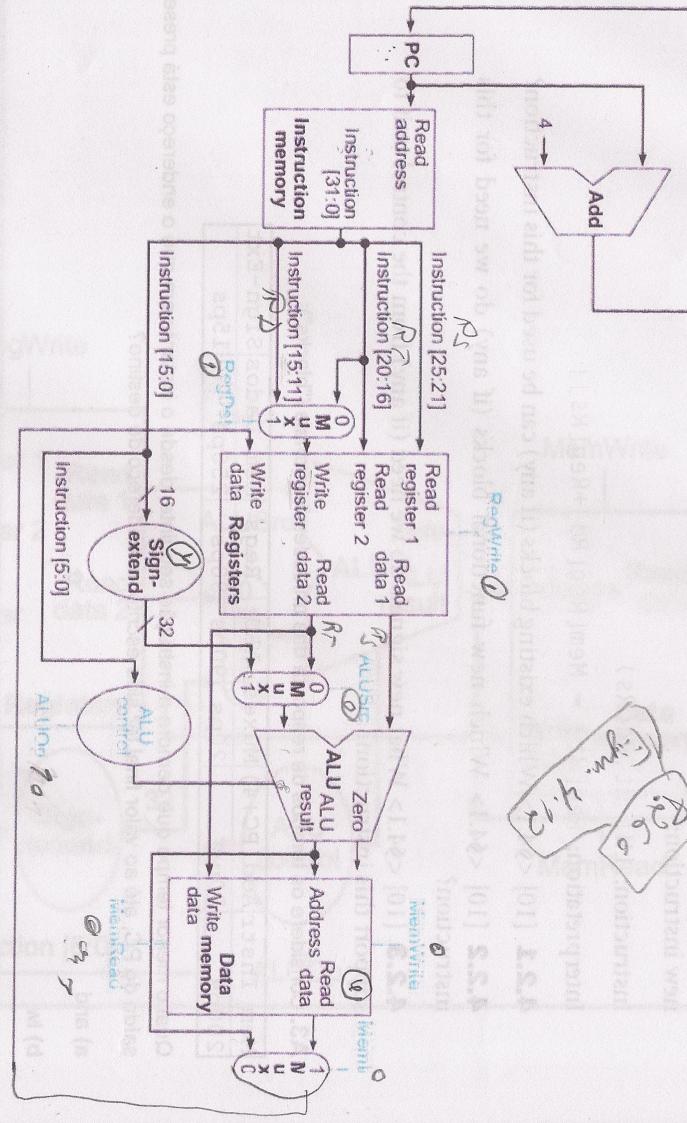
Blue 0000FF

Red FF0000

Grey 808080

Considerando as seguintes figuras / 1 2 3 que lhe manda

11



4.1 Consider the following instruction:

Instruction: AND Rd, R_s, R_t

Interpretation: $\text{RegLKa}_j = \text{RegLRs}_j$ AND RegL Rt_j

4.1.1 [5] <§4.1> What are the values of control signals generated by the control instruction Figure 4.2 for the above instruction? $R_{00} = 1 \quad R_{01} = 1 \quad R_{10} = 0 \quad R_{11} = 0$

4.1.2 [5] <§4.1> Which resources (blocks) perform a useful function for this instruction? 10

4.1.3 [10] <§4.1> Which resources (blocks) produce outputs, but their outputs are not used for this instruction? Which resources produce no outputs for this instruction?

4.1.X. Repita o exercício para as instruções

- a) lw** $R T, iimm(RS) \# RT = sign_ext[imm_w[RS + sign_extend(immm)]]$

b) sw $RT, imm(RS) \# mem_w[RS + sign_extend(immm)] = RT$

4.2 The basic single-cycle MIPS implementation in [Figure 4.2](#) can only implement some instructions. New instructions can be added to an existing Instruction Set Architecture (ISA), but the decision whether or not to do that depends, among other things, on the cost and complexity the proposed addition introduces into the processor datapath and control. The first three problems in this exercise refer to the new instruction:

Instruction: $LW1\ Rt, Rd(R_S)$

Interpretation: $Reg[Rt] = Mem[Reg[Rd] + Reg[R_S]]$

4.2.1 [10] $<\$4.1>$ Which existing blocks (if any) can be used for this instruction?

4.2.2 [10] $<\$4.1>$ Which new functional blocks (if any) do we need for this instruction?

4.2.3 [10] $<\$4.1>$ What new signals do we need (if any) from the control unit to support this instruction?

A3. Consider the response times of functional units ("blocks"):

Mem	Instr Add (PC+4)	Muxes	ALU	Regs	Mem Dados	Sign-Ext
200ps	70ps	20ps	90ps	80ps	250ps	15ps

Qual o menor tempo que demora a instrução seguinte, desde o instante em que o endereço está presente na saída do PC, até ao valor final ter sido escrito no endereço de destino?

- a) and
- b) lw

Última alteração: Segunda, 29 Setembro 2014, 08:44

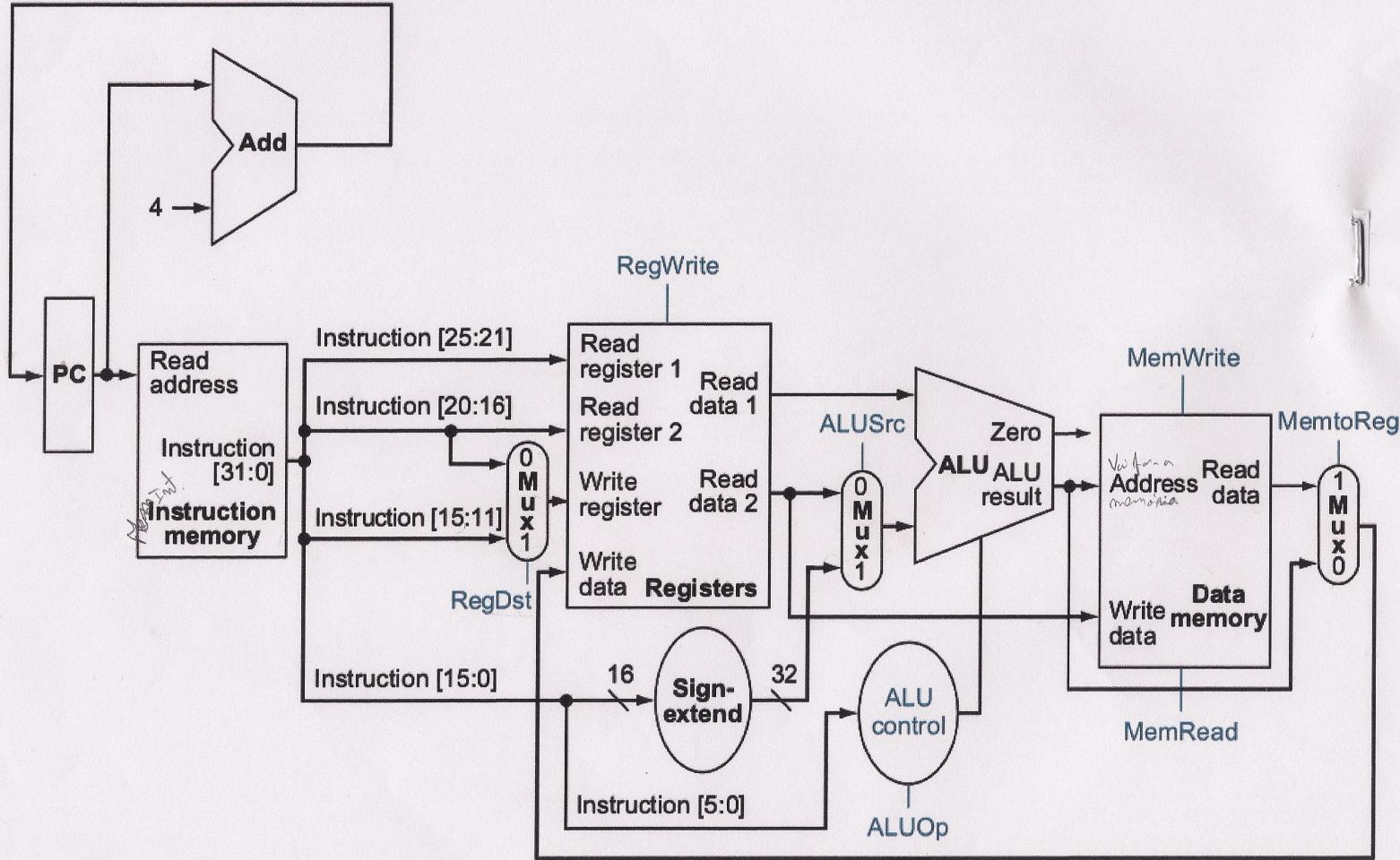
18. set. de 08:44 (Má modificações)

ultimo set. em 29 Setembro 2014, às 08:44 por "má modificações".
Apenas o bloco de memória é que não pode ser usado, mas o bloco de ALU pode ser usado.

é só trocar o bloco de ALU pelo bloco de memória, e trocar o bloco de memória pelo bloco de ALU.
é só trocar o bloco de ALU pelo bloco de memória, e trocar o bloco de memória pelo bloco de ALU.

é só trocar o bloco de ALU pelo bloco de memória, e trocar o bloco de memória pelo bloco de ALU.

"Uma solução para obter o bloco de memória é usar (we_M1 brisa) e o bloco de ALU é usar (Y1_A1



4.2.x) Indique todos os sinais de controlo

4.2.y) repetir os anteriores com a instrução seq rd, rs, rt (que coloca o valor 1 no registo rd se os valores em rs e rt forem iguais, e 0 se forem diferentes), sem modificar a ALU

A3. Considere os tempos de resposta das unidades funcionais ("blocks"):

Mem Instr	Add(PC+4)	Muxes	ALU	Regs	Mem Dados	Sign-Ext
200ps	70ps	20ps	90ps	80ps	250ps	15ps

Qual o menor tempo que demora a instrução seguinte, desde o instante em que o endereço está presente à saída do PC, até ao valor final ter sido escrito no endereço de destino?

a) and

b) lw

Última alteração: Segunda, 12 Outubro 2015, 19:55

4.1.1

QW ma today

$$\begin{array}{l} \text{Reg WdR} = 1 \quad / \quad \text{ALU SRC} = 1 \quad / \quad \text{Mem WdR} = 0 \\ \text{Reg DST} = 0 \quad / \quad \text{ALU OF} = 00 \quad / \quad \text{Mem TO Reg} = 1 \\ \text{Mem Read} = 1 \end{array}$$

4.1.1

SW

ma ma

(e)

$$\begin{array}{l} \text{Reg WdR} = 0 \quad / \quad \text{ALU SRC} = 1 \quad / \quad \text{Mem WdR} = -1 \\ \text{Reg DST} = 1 \quad / \quad \text{ALU OF} = 00 \quad / \quad \text{Mem TO Reg} = 0 \\ \text{Mem Read} = 0 \end{array}$$

Avda 13.1

lendimeto
Con objeto
Reg. gen.
white

	Mem Read	ALU Sre	Mem PWhite	Cont. de ALU	Alu of	Mem to Reg	Reg DST
AND	① no AND Vas hacer ANDita	a)	0	AND 10000	0	1	
SW	0	0	1	ADD 10010	X indiferencia	X indiferencia	
LW	1	1	1	ADD 0010	1	0	

AND o sum. inteiros aritméticos

SW Mem to reg - o sum. indiferencia, faz mais coisas e escreve mais na memória mem 0 mem 1

Res DST - " " " " " " " " tambem nos saido uns 1s
para escrever algo

SEQ	1	0	0	0	Sub 0110	X indiferencia	①
LWI	1	1	1	0	Cont. de operações	1	0

Entendendo implementação
Seq = SFT Compara 2 registo
se forem iguais devolve valor 1
mete 1

$$\text{Seq } T_0, T_1, T_2 \rightarrow \begin{cases} T_1 = T_2 \\ T_0 = 1 \end{cases}$$

$$\begin{cases} T_1 \neq T_2 \\ T_0 = 0 \end{cases}$$

4.2

elemento de datos

elemento de control

elemento de memoria

AND: Toda excepto:
Mem, sign extendSign extend
ignorado

Mem

SW: Toda excepto:
MUX Mem to Reg
Reg Dst

Mem

Por un ALU SR es necesario no tener mas 1 bit sign extendido, por lo que el sign extendido es ignorado.

Mem Instruc	Add (Pc+4)	Muxer	ALU	Regs	Mem Write Mem dados	Sign extend
AND	200 ps	70 ps	20 fs	90	80	250

AND	Mem read ① 200	20 fs ③ 20 fs ⑤	90 ④	80 ① ② 80 ⑥		= 490 fs
SW	200 ①	20 ⑤	90 ③	80 ② 80 ⑥	250 ④	= 720 fs

Exercice 4.3-1

Load Word Instructions mire lente \rightarrow LW

I-Mem	Add	Mux	ALU	Regs	D-Ram	Control blocks	
400	100	30	120	200	350	100	
LW	400 ①	30 ⑤	120 ③	200 ⑥	350 ④		= 1300 Temp. ^{do cicles} do relogio , normal

Com o Improvement (350) da ALU 120 para a 420 //

Logico improvement 1600 tempo de ciclo do relogio com melhoria (Improvement)

4.3-2

$$\text{Speedup} = \frac{1 \times 1300}{0,95 \times 1600} = 0,86$$

Logro depois da melhoria de 5%
 o ~~tempo de ciclo do relogio é menor~~ ^{máx}
 a melhoria de 0,14% do Speedup
 o resultado do Speedup = 0,86 e' inferior
 a 1 logro e' uma ~~melhoria~~ menor de
 melhoria de 5%

4.7

Agregado 54

SW

901011 00011100010 / 000000000000 0010100

Op. Pato -

31 ~~+~~ 0

[5 - 0]

4.7.1

Sign extend: 000000000010100 = 2⁰₍₁₀₎

shift left 2 : 0000000001010000

4.7.2

Instruction [05-0] Func = 010100

ALU op: ~~funcion add~~ mca of instructions 5W logo ALU op = 00
Add

F_{control} =

4.7.3

PC + 4

instruc: (20-16)

Reg. Mux: 00010 ALU Mux: sign extend 2⁰₍₁₀₎

Mem. Mux: Don't care de Mem To Reg

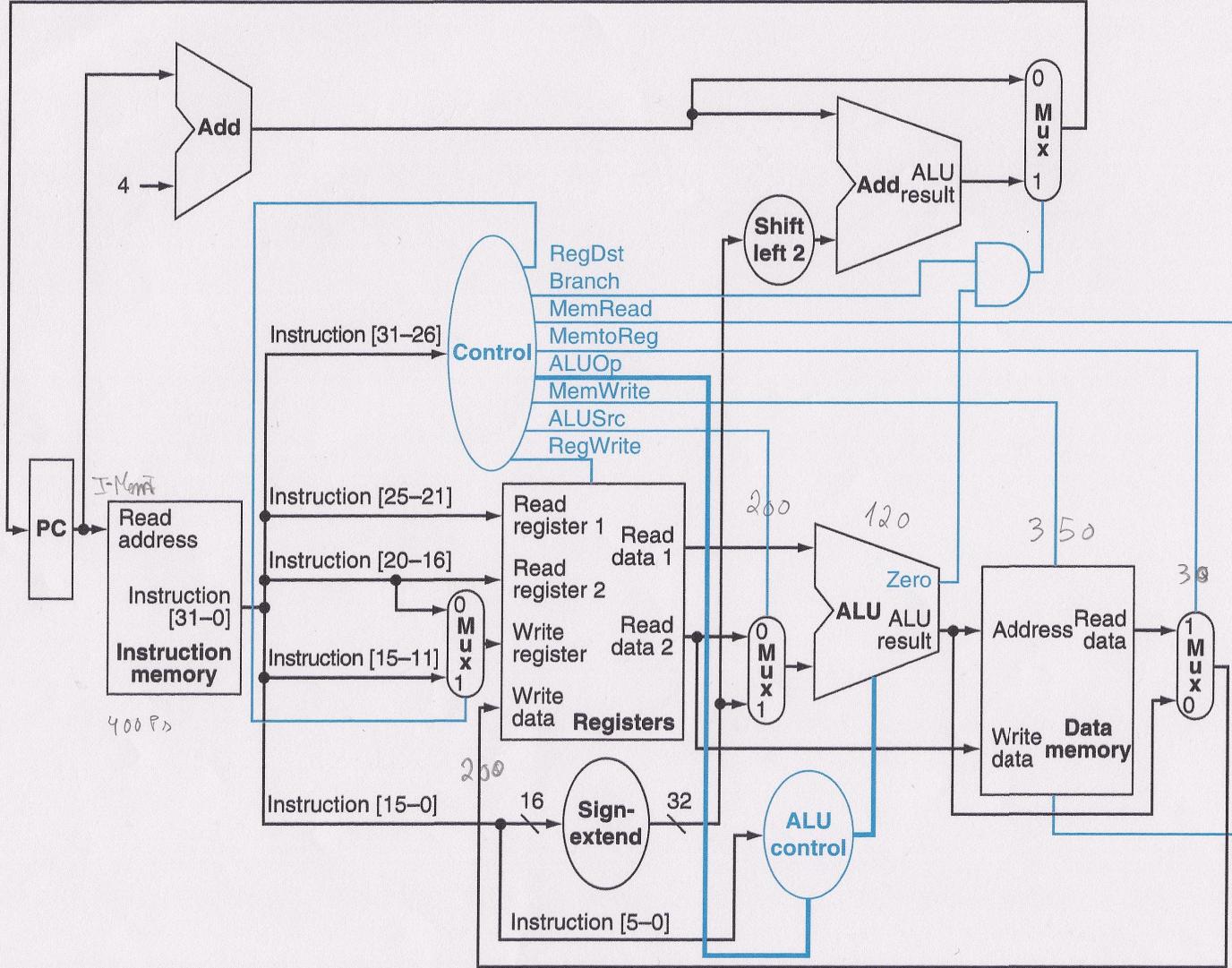
Branch Mux: PC + 4

on Nai o Contenido de memoria

on Nai o Contenido de dato

LW -> instructions + data

4.3.1
4.3.2



4.7.4

